VistALink 1.5

Release Notes

May 2006

Application Modernization Program
Health Systems Design & Development (HSD&D)
Department of Veterans Affairs

TABLE OF CONTENTS

Introduction

VistALink is a transport layer that provides communication between HealthVet-VistA Java applications and VistA/M servers, in both client-server and n-tier environments. It allows Java applications to execute remote procedure calls (RPCs) on the VistA/M system and retrieve results, synchronously. VistALink 1.5 is also referred to as "VistALink J2M."

VistALink J2M consists of two parts:

- *Java-side adapter libraries* – these use the J2EE Connector Architecture (J2CA) 1.0 specification to integrate J2EE and Java applications with legacy systems, such as Vista/M systems.
- *M listener process* – receives and processes requests from client applications.

New Features and Functions Added to the Software

J2EE Connector Architecture (J2CA)-Compliant Resource Adapter

The first version of VistALink (1.0) supported Java rich clients connecting to M servers, but did not support connectivity from a J2EE middle tier. VistALink 1.5 addresses this need, by adding support for the J2EE Connector Architecture (J2CA), a J2EE standard for integrating J2EE applications with non-J2EE enterprise information systems.

VistALink J2M now provides a J2CA-compliant resource adapter, enabling J2EE-based middle-tier application code to communicate with VistA/M systems. By complying with the J2CA specification, the adapter is able to take care of built-in J2EE platform support for J2CA, including connection pooling.

The VistALink resource adapter is packaged in a Resource Adapter Archive (RAR), in both exploded and packaged formats.

J2EE Connector Proxy User Security

The *connector proxy user* is the M server's primary gatekeeper for controlling access to M from the new VistALink J2CA resource adapter. This is a special user class in the Kernel NEW PERSON file (#200). Only J2EE connectors configured by IRMs with credentials for "connector proxy" Kernel users can connect to an M VistALink listener. Once a connection is established through a connector proxy user, J2EE applications on the J2EE server can execute a variety of RPCs on the M server under a variety of end-user identities.

VistALink leverages the connector proxy functionality and security features of Kernel patch XU*8.0*361. The Foundations menu has a new menu option to create a connector proxy user. The menu option wraps the CONT^XUSAP entry point.

J2CA Re-Authentication Connection Specification Types

When using the VistALink J2CA connector, it is the J2EE application's responsibility to specify (re-authenticate) the end-user identity with which to execute each remote procedure call (RPC) on the M system.

Several classes in the `gov.va.med.vistalink.adapter.cci package` support different ways for the J2EE application to identify (re-authenticate) the end user to the M system:

- VistaLinkDuzConnectionSpec (if application knows the user's DUZ, e.g. via logon)
- VistaLinkVpidConnectionSpec (if application knows the user's VPID, e.g. via logon)
- VistaLinkAppProxyConnectionSpec (see *Application Proxy Users*, below)

Application Proxy Users

`VistaLinkAppProxyConnectionSpec` allows the J2EE application to execute RPCs on the M system under the identity of a special application-level *application proxy user*, rather than the actual end-user. Application proxies are expected to be used in either of the following special situations:

- The J2EE end-user does not have a user account on the M system on which an RPC is to be executed, i.e., when a service uses VistALink from an EJB (when an end-user is not practical)
- It is not appropriate for the RPC to execute under the identity of a particular end-user.

Kernel patch XU*8*361 is required to support the new functionality.

Note: A "tester" application proxy user is distributed with VistALink. It is added to the NEW PERSON file (#200) during the installation post-init, and named “XOBVTESTER,APPLICATION PROXY.”

Institution Mapping API

J2EE applications retrieve resource adapters from the Java Naming and Directory Interface (JNDI), using a JNDI name. Applications using VistALink J2CA resource adapters usually need a way to retrieve the resource adapter associated with a particular M system. This is because multiple VistALink resource adapters, connecting to different M systems, can be configured on a given J2EE system.

VistALink's institution mapping functionality allows J2EE applications to retrieve the JNDI name of a particular resource adapter, based on the station number of the M system it wishes to connect to. The station number is a piece of information any multi-divisional-aware application is likely to possess. The institution mapping API is provided in the `gov.va.med.vistalink.institution.InstitutionMappingDelegate` class.

VistaLinkRequestRetryStrategy API

While executing an RPC, connectivity to the VistA system can fail, due to a network failure or anything else that breaks connectivity. For J2EE applications using the VistALink J2CA resource adapter, the application request’s implementation of the `VistaLinkRequestRetryStrategy` interface determines whether or not VistALink automatically retries the request (by retrieving a new connection from the resource adapter connection pool).

Two default retry strategies are provided: `VistaLinkRequestRetryStrategyAllow` and `VistaLinkRequestRetryStrategyDeny`. The strategy interface is `gov.va.med.vistalink.adapter.record.VistaLinkRequestRetryStrategy`.

Environment API

The `gov.va.med.environment.Environment` API exposes the following environment settings on the J2EE server:

- `isProduction` – this J2EE setting must match the production/test setting of the M system
- `getServerType` – currently, this setting is

  JBOSS | ORACLE | SUN_RI_13 | UNKNOWN | WEBLOGIC | JAVASE | WEBSPHERE

The values returned by these APIs are currently controlled by JVM arguments configured by the J2EE system manager.

WebLogic Console Plug-in to Monitor Resource Adapter

VistALink 1.5 provides a console plug-in for the WebLogic administration console. The VistALink console provides visibility into the internal functioning of VistALink connectors and helps with some VistALink management tasks. The VistALink console runs as an extension in the WebLogic console, on the administration server for a given WebLogic configuration.

The VistALink console provides both summary and detailed views of deployed VistALink resource adapters, as well as the current institution mapping values. The detailed connector view can be used to test the connectivity of the resource adapter, because accessing the detailed view makes calls over the adapter to the corresponding Vista/M system to retrieve information used in the detailed display.

J2EE Configuration File Editor

Another part of the WebLogic console plug-in is the VistALink configuration editor. It provides a user-friendly interface for editing the VistALink configuration file, which persists configuration settings for each VistALink adapter/connector.

J2EE Sample Application

The new VistALink sample Web application demonstrates the use of the VistALink J2CA resource adapter from a J2EE middle tier. It demonstrates use of all of the connection specification types, including `VistaLinkAppProxyConnectionSpec`.

Log4J Logger Listing

VistALink core log4j categories are documented in the spreadsheet provided in the **log4j** subdirectory of the VistALink distribution file. Each logger category provides a description and notes the supported logger levels (e.g., DEBUG and ERROR). You can use the logger category name(s) to set up loggers in your log4j configuration files, as needed, to log information from specific parts of the VistALink package.

New M Site Parameters

Two new M site parameters are provided for supporting J2EE connectivity:

- **J2EE Connection Timeout**: This field indicates the number of seconds that a VistALink connection from J2EE to M should be allowed to remain connected when inactive, before M drops the connection.
- **J2EE Reauthentication Timeout**: This field indicates the number of seconds between 180 and 3600 (inclusive) before a re-authenticated connection is considered expired and must go through the re-authentication process again (default: 600 secs).

Site parameters can be edited using the Site Parameters action in the Foundations menu.

Enhancements and Modifications to Existing Software

Modified VLINK.COM files

VistALink M listeners are typically configured as TCP/IP services on VAMC systems, as per HSITES guidance. VistALink **VMS .COM** files will need to be updated, similar to the modifications now required for Broker, MailMan, and HL7 services.

HSITES has provided a cookbook (VISTALINK_TCPIP_COOKBOOK.DOC) and **.COM** files to assist VistA/M system managers who need to create the VLINK service or simply modify the existing VLINK service files. For more information, please see the VistALink Installation Guide.

Specify Any Host/Port in SwingTester Application

The J2SE sample/test application (`gov.va.med.vistalink.samples.VistaLinkRpcSwingTester`) now allows ad hoc entry of the IP and port of the target VistALink listener. Previously, an explicit configuration was required in the **jaas.config** file to connect to a VistALink listener.

CCOW-Based SSO/UC Support

VistALink's client/server mode now supports single sign-on to M systems based on CCOW user context, as implemented by the Office of Information's SSO/UC (Single Sign-On/User Context) project.

For application developers, a new CCOW-enabled JAAS (Java Authentication and Authorization Service) `CallbackHandler` class (`CallbackHandlerSwingCCOW)` is provided for the `VistaLoginModule` class. A new sample application (`gov.va.med.vistalink.samples.VistaLinkRpcSwingSimpleCcow)` is also provided, to demonstrate the coding necessary to implement CCOW-based SSO/UC support.